UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/725,728	12/02/2003	Kwasi Addo Asare	RSW920030191US1 (123)	3074

46320   7590   09/11/2009
CAREY, RODRIGUEZ, GREENBERG & PAUL, LLP
STEVEN M. GREENBERG
950 PENINSULA CORPORATE CIRCLE
SUITE 3020
BOCA RATON, FL 33487

EXAMINER
BROPHY, MATTHEW J

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
09/11/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/725,728
Filing Date: December 02, 2003
Appellant(s): ASARE ET AL.

Scott Paul
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed May 19, 2009, appealing from the Office

action mailed December 19, 2009.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is substantially correct. The changes are as follows:

WITHDRAWN REJECTIONS

1.   The following grounds of rejection are not presented for review on appeal because they have been withdrawn by the examiner.

a.   Claims 1-12 rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Applicant has stated on the record in the Appeal Brief that "a repository is

a storage device…script generation must necessarily be tied to hardware"

and will be interpreted as such from here forward.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

6,681,391	Marino et al.	01-2004
2004/0078793	Bragulla	4-2004

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

2.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

3.   Claims 1-17 are rejected under 35 U.S.C. 103(a) as being unpatentable over

USPN 6,681,391 Marino et al. in view of US PG Publication 2004/0078793 Bragulla et

al.

        Regarding Claims 1 and 5, Marino teaches: a repository of semantic models

(230) for interdependent ones of application components **(222) (Col. 13, Ln 19-35**

"Additional information may be received by the installation-affecting-relationship

manager 220 from a Core Engine 224, e.g., a list of components 222 to be

installed and the like. The installation-affecting-relationship manager 220

computes entries for an installation-relationship matrix 226 and initializes

parameters 228 and in a data structure 230 as well as an Installation_order list

232 (step 202).");

 a mapping of individual listings **(translation of 230 to 232, FIG.5, Col. 9, Ln 40-43**

"The graph [stored in the matrix as described] corresponding to the dependency

relationships can be used to perform a depth first tree traversal and generate a

valid installation order, if one exists" See further Col. 10, Ln 64-68: "Following an

examination of the matrix in table 2, the component A3 is selected and placed in

an Installation order list because i[t] has an in-degree of 0 and has the largest

index among the nodes with in-degree of 0.") in said semantic models to target

platform specific installation instructions ("specific" taught by update of 232 based

on user's platform-Col. 14, Ln 13-31 "The current state of the user's machine is

determined. If one or more components are incompatible with the components to

be loaded, the offending components are removed in an orderly fashion so that

components dependent on them are also removed. The procedure described for

the mutually exclusive relationship can be used to implement this operation in

accordance with the invention. This is followed by installation of the requested

components, also in accordance with the invention, e.g., the method outlined in

the flow chart in FIG. 4.");

and, a <script> generation engine configured to produce a target specific set of

instructions (232) for a specified application component based upon a mapping

(translation of 230 to 232, FIG.5, Col. 9, Ln 40-43- "The graph [stored in the matrix

as described] corresponding to the dependency relationships can be used to

perform a depth first tree traversal and generate a valid installation order, if one

exists" See further Col. 10, Ln 64-68: "Following an examination of the matrix in

table 2, the component A3 is selected and placed in an Installation order list

because i[t] has an in-degree of 0 and has the largest index among the nodes with

in-degree of 0.") of at least one of said semantic models in said repository (Col. 14,

Ln 13-31 "The current state of the user's machine is determined. If one or more

components are incompatible with the components to be loaded, the offending

components are removed in an orderly fashion so that components dependent on

them are also removed. The procedure described for the mutually exclusive

relationship can be used to implement this operation in accordance with the

invention. This is followed by installation of the requested components, also in

accordance with the invention, e.g., the method outlined in the flow chart in FIG.

4.")

Marino does not explicitly teach a "script" generation engine. However, this limitation is

taught by Bragulla: ("¶[0031] The inventiveness is particularly evident in that an

installation script is generated automatically and above all dynamically; the new

package is augmented with specific target system information, so that optimized

and in particular time-optimized installation on the relevant target system is

possible without further system parameters having to be requested and/or

processed during the installation. The time optimization of the status change,

particularly of software replacement, is therefore based according to the

invention on the concept that the software to be installed does not need to be

moved during the active phase.") In addition, it would have been obvious to one of

ordinary skill in the art to combine the teachings of Bragulla with the teachings of Marino

as Marino describes the advantages of using a script for installation, and Bragulla

teaches a method for overcoming the limitations described in Marino (automatic script

customization): Col. 1 Ln 67-Col. 2 Ln 6 "A single installation script may be used to

install the program, and, for each individual computing environment the script can be
edited to account for the peculiar requirements of that context. The disadvantage of this
method is that it requires an administrator to spend a considerable amount of time
editing each individual script to tailor it to particular configurations."

Regarding Claims 2 and 6, Marino teaches: wherein each of said semantic
models comprises a listing of component relationships **(226)**, target platform
requirements **("The current state of the user's machine is determined.")** and
platform neutral installation instructions **(232 previous to update for user's platform)**.
**(Col. 14, Ln 13-31 "The current state of the user's machine is determined. If one or
more components are incompatible with the components to be loaded, the
offending components are removed in an orderly fashion so that components
dependent on them are also removed. The procedure described for the mutually
exclusive relationship can be used to implement this operation in accordance
with the invention. This is followed by installation of the requested components,
also in accordance with the invention, e.g., the method outlined in the flow chart
in FIG. 4.").**

Regarding Claims 3 and 7, Marino teaches: wherein said component
relationships comprises at least one component relationship selected from the group
consisting of a containment relationship, a usage relationship, a contradiction
relationship, and an equivalence relationship. **(Col. 2, Ln 52-65 "Briefly, the invention
requires that a suite, and preferably each component of the suite, includes
specification of its installation-affecting relationships along with an acceptable**

installer, in particular if the installer is different from a default installer. Some
examples of installation-affecting relationships include dependency or mutual-
exclusion. In a dependency relationship a first component depends on a second
component if the second component has to be present prior to the installation of
the first component. In contrast, a mutual-exclusion relationship between two
components requires that only one of them can be present in a computing
environment, or a sub-part of a computing environment, thus requiring the
removal of one, if present, to install the other.")

Regarding Claim 4, Marino teaches: further comprising a Web services interface
to said repository configured to permit remote access to said repository. **((2) "... The**
invention may also be practiced in distributed computing environments where
tasks are performed by remote processing devices that are linked through a
communications network..." and (14) "Software may be downloaded, typically in
compressed files, over networked computers, including those connected to the
Internet.")

Regarding Claims 8 and 13 Marino teaches: retrieving a semantic model for the
application component from a communicatively coupled repository of semantic models
(Col. 13, Ln 19-35 "Additional information may be received by the installation-
affecting-relationship manager 220 from a Core Engine 224, e.g., a list of
components 222 to be installed and the like. The installation-affecting-
relationship manager 220 computes entries for an installation-relationship matrix

226 and initializes parameters 228 and in a data structure 230 as well as an Installation_order list 232 (step 202).");

determining from said semantic model **(230, Matrix entries 226, FIG. 5)**, a set of dependent components required to be present in the specific target platform **(232, see below Col. 14, Ln 13-31) (Col. 13, Ln 19-35 "Additional information may be received by the installation-affecting-relationship manager 220 from a Core Engine 224, e.g., a list of components 222 to be installed and the like. The installation-affecting-relationship manager 220 computes entries for an installation-relationship matrix 226 and initializes parameters 228 and in a data structure 230 as well as an Installation_order list 232 (step 202).")**

further determining from said semantic model a set of resource requirements required to be met by the specific target platform; **(Col. 14, Ln 13-31 "The current state of the user's machine is determined…)**

and, mapping said set of dependent components and said set of resource requirements into platform specific instructions **(Col. 14, Ln 13-31 "If one or more components are incompatible with the components to be loaded, the offending components are removed in an orderly fashion so that components dependent on them are also removed. The procedure described for the mutually exclusive relationship can be used to implement this operation in accordance with the invention. This is followed by installation of the requested components, also in accordance with the invention, e.g., the method outlined in the flow chart in FIG. 4.");**

Marino does not explicitly teach: in a platform specific installation script. However, this

limitation is taught by Bragulla: (**"¶[0031] The inventiveness is particularly evident in**

that an installation script is generated automatically and above all dynamically;

the new package is augmented with specific target system information, so that

optimized and in particular time-optimized installation on the relevant target

system is possible without further system parameters having to be requested

and/or processed during the installation. The time optimization of the status

change, particularly of software replacement, is therefore based according to the

invention on the concept that the software to be installed does not need to be

moved during the active phase.") In addition, it would have been obvious to one of

ordinary skill in the art to combine the teachings of Bragulla with the teachings of Marino

as Marino describes the advantages of using a script for installation, and Bragulla

teaches a method for overcoming the limitations described in Marino (automatic script

customization): Col. 1 Ln 67-Col. 2 Ln 6 "A single installation script may be used to

install the program, and, for each individual computing environment the script can be

edited to account for the peculiar requirements of that context. The disadvantage of this

method is that it requires an administrator to spend a considerable amount of time

editing each individual script to tailor it to particular configurations."

Regarding Claims 9 and 14, Marino teaches: yet further determining from said

semantic model a set of platform neutral installation operations **(Col. 13, Ln 19-35**

"Additional information may be received by the installation-affecting-relationship

manager 220 from a Core Engine 224, e.g., a list of components 222 to be

installed and the like. **The installation-affecting-relationship manager 220
computes entries for an installation-relationship matrix 226 and initializes
parameters 228 and in a data structure 230 as well as an Installation_order list
232 (step 202).");** and, further mapping said set of platform neutral installation
operations into said platform specific instructions. **(Col. 14, Ln 13-31 "The current
state of the user's machine is determined. If one or more components are
incompatible with the components to be loaded, the offending components are
removed in an orderly fashion so that components dependent on them are also
removed. The procedure described for the mutually exclusive relationship can be
used to implement this operation in accordance with the invention. This is
followed by installation of the requested components, also in accordance with the
invention, e.g., the method outlined in the flow chart in FIG. 4.").**

Regarding Claims 10 and 15, Marino teaches: identifying a set of dependent
components for the application component; and, further identifying a set of sub-
dependent components for at least one of said dependent components. **(Col. 2, Ln 52-
65 "Briefly, the invention requires that a suite, and preferably each component of
the suite, includes specification of its installation-affecting relationships along
with an acceptable installer, in particular if the installer is different from a default
installer. Some examples of installation-affecting relationships include
dependency or mutual-exclusion. In a dependency relationship a first component
depends on a second component if the second component has to be present
prior to the installation of the first component. In contrast, a mutual-exclusion**

relationship between two components requires that only one of them can be
present in a computing environment, or a sub-part of a computing environment,
thus requiring the removal of one, if present, to install the other.").

Regarding Claims 11 and 16, Marino teaches: repeating the identifying and
further identifying steps for each dependent and sub-dependent component in a
hierarchy of dependent components for the application component. **(Col. 2, Ln 52-65**
"Briefly, the invention requires that a suite, and preferably each component of the
suite, includes specification of its installation-affecting relationships along with
an acceptable installer, in particular if the installer is different from a default
installer. Some examples of installation-affecting relationships include
dependency or mutual-exclusion. In a dependency relationship a first component
depends on a second component if the second component has to be present
prior to the installation of the first component. In contrast, a mutual-exclusion
relationship between two components requires that only one of them can be
present in a computing environment, or a sub-part of a computing environment,
thus requiring the removal of one, if present, to install the other.")

Regarding Claims 12 and 17, Marino further teaches: computing a composite set
of resource requirements for the application component and for said set of dependent
components. **(Col. 13, Ln 19-35 "The installation-affecting-relationship manager**
220 computes an in-degree parameter and a scope parameter for each
component 222 to be installed or processed (step 204).")

(10) Response to Argument

In Appeal Brief, Appellant Argues:
Claim 1
Independent claim 1 recites "a repository of semantic models for interdependent ones of application components." On page 3 of the Fourth Office Action, the Examiner asserted that the data structure 230 of Marino corresponds to the claimed repository of semantic models and the "components 222 to be installed" of Marino corresponds to the claimed "interdependent ones of application components." Appellants respectfully submit that the Examiner has mischaracterized the scope and content of Marino.

Notably absent from the Examiner's analysis is a claim construction for the terms at issue. For example, the claimed "a repository of semantic models" requires a plurality of semantic models being stored with a repository (i.e., a storage device). This also implies that the semantic models were previously created (in order to be stored). Column 13, lines 19-35 of Marino, which is the passage cited by the Examiner to teach these limitations, does not teach all of these limitations. Instead, Marino discloses that the "installation-affecting-relationship manager 220 computes entries for an installation-relationship matrix 226 and initializes parameters 228," which are found within the data structure 230 (see Fig. 5), and these calculations appear to be done contemporaneously upon receiving information about the components to be installed.

Notably absent from these passages is any indication that a plurality of data structures 230 (i.e., allegedly corresponds to the claimed semantic models) are stored in a repository. The absence of such a teaching is consistent with Marino, which teaches that the information contained within the data structure 230 is computer in an as-needed manner.

Examiner's Response:

Examiner respectfully disagrees. First, it should be noted, the claim does not recite a

"plurality of semantic models" but rather a repository or semantic models. Regardless,

The data structure 230 includes an "Installation-Relationship Matrix 226", which holds a

plurality of entries which model the dependencies between the application components.

For example, Col. 10, Ln 44, States "Matrix Table 2, corresponds to the equivalent

graphs 130 and 132 in FIG. 3, which are termed digraphs or directed graphs." That's is,

there is one data structure 230, that the examiner equates to the repository, which in

turn holds a matrix of multiple dependency entries which equates the semantic models

of the applicant's claim. Therefore, this rejection is maintained.

In the Appeal Brief Appellant Argues:

> Additionally, although the claimed specifically recites "semantic models for
> interdependent ones of application components" (emphasis added), column
> 13, lines 33-35 of Marino specifically states the following: "there is no
> requirement that components 222 be related or be represented as a single
> tree or connected graph." Thus, for the above presented reasons," Marino
> fails to teach the limitations for which the Examiner is relying upon Marino
> to teach.

Examiner's Response:

Examiner respectfully disagrees. The "semantic models for interdependent ones of

application components" are anticipated by Marino's Installation-Relationship Matrix

226, part of 230, which is a sparse matrix of dependencies between the components

(Column 7, Ln 35-51, See E.g. Tables 1 and 2). While Marino states as applicant has

quoted, that "there is no requirement that components 222 be related" this is unrelated

to the teaching of the matrix of dependencies. In fact, as a sparse matrix, many of the

components are not dependent on each other, however, the *matrix does represent the*

dependencies that do exist. Marino essentially states that there's no requirement that

each component is related to each other component, but this does not negate the fact

that Marino teaches an "Installation-<u>Relationship</u> Matrix". Further, there is nothing in the

claim to require a) that all components must be interdependent; b) non-interdependent

components might be represented in the repository. Therefore, Applicant's argument is

unrelated to the claim language, or the pertinent teachings of the prior art, and are

therefore unpersuasive.

In Appeal Brief Appellant Argues:

> Independent claim 1 further recites "a mapping of individual listings in said
> semantic models to target platform specific installation instructions." With
> regard to these limitations, the Examiner asserted the following in the
> paragraph spanning pages 3 and 4 of the Fourth Office Action: a mapping
> of individual listings (translation of 230 to 232, FIG.5) in said semantic
> models to target platform specific installation instructions ("specific" taught
> by update of 232 based on user's platform) (Col. 14, Ln 13-31 "The current
> state of the user's machine is determined. If one or more components are
> incompatible with the components to be loaded, the offending components
> are removed in an orderly fashion so that components dependent on them
> are also removed. The procedure described for the mutually exclusive
> relationship can be used to implement this operation in accordance with the
> invention. This is followed by installation of the requested components, also
> in accordance with the invention, e.g., the method outlined
> in the flow chart in FIG. 4.") (emphasis in original)

> Although the Examiner refers to "translation of 230 to 232," no such
> translation exists. Instead, column 13, lines 26-30 of Marino states: The
> installation-affecting-relationship manager 220 computes entries for an
> installation-relationship matrix 226 and initializes parameters 228 and in a
> data structure 230 as well as an Installation_order list 232 (step 202).

> Thus, there is no "translation of 230 to 232" since the manager 220
> appears to independently create the data structure 230 and the installation
> order list 232. As such, the installation order list 232 is not mapped to
> anything (i.e., individual listings) within the alleged semantic model (i.e.,
> data structure 230).

Examiner's Response:

Examiner respectfully disagrees. The translation of the information in 230 to the

installation order list 232 is evidenced by the discussion of the generation of the

installation order list in Columns 7-9 and evidenced further by the two arrow to the data

structure 230 in FIG. 5. Specifically, while the generation of the data structure 230 and

installation order list 232 is described in the same step (202), Marino's disclosure

teaches the generation of that list is accomplished specifically, by traversing the

relationship graphs represented in the relationship matrix 226:

(Col. 9, Ln 40-43) "The graph [stored in the matrix as described] corresponding to the

dependency relationships can be used to perform a depth first tree traversal and

generate a valid installation order, if one exists" See further Col. 10, Ln 64-68:

"Following an examination of the matrix in table 2, the component A3 is selected and

placed in an Installation order list because i[t] has an in-degree of 0 and has the largest

index among the nodes with in-degree of 0.")

In Appeal Brief, Appellant Argues:

> With regard to the "target platform specific installation instructions," the
> Examiner's cited passage of column 14, lines 13-31 is not relevant to the
> claimed limitations. As claimed, certain individual listings (which are found
> in the semantic models) are mapped to a target platform specific
> installation instructions. As such, when a specific target platform is
> encountered, the requirements (e.g., installation instructions) for the
> specific target platform can be identified (see discussion regarding block
> 310 in paragraph [0029] and discussion of feature 210 within paragraph
> [0026] of Appellants' disclosure). Although the Examiner's cited passage
> refers to the "current state of the user's machine," this state is not related

to the target platform. Instead, the state refers to the components (i.e., applications that run on a computing platform). As one having ordinary skill in the art would recognize, there is a fundamental difference between the components (which run on a computing platform) and the platform itself.

Claim 2

Dependent claim 2 recites "each of said semantic models comprises a listing of component relationships, target platform requirements and platform neutral installation instructions." Regarding the claimed target platform requirements, on page 5 of the Fourth Office Action, the Examiner referred to the teaching of "[t]he current state of the user's machine is determined." Appellants' position is that the current state of a user's machine is not comparable to the claimed target platform requirements. The "current state" of a machine merely reflects the current configuration of the user's machine. On the contrary, the target platform requirements, describe the requirements of the target platform. These are entirely different concepts. Thus, the Examiner has mischaracterized the scope and content of the applied prior art with regard to claim 2.

Examiner's Response:

Examiner respectfully disagrees. Examiner cited the current state determination for the

user's computer, because this determination which components are compatible with the

specific user's computer (i.e. target platform). From this determination, it calculates the

installation list (installation instructions) to be compatible with the specific user computer

system requirements. See Col. 14, Ln 13-17, "In response to a request for an

application to be launched, the core installation engine 224 and the installation-

affecting-relationship manager 220 are invoked in the context of the requested

application, i.e., its components and sub-components. The current state of the user's

machine is determined. If one or more components are incompatible with the

components to be loaded, the offending components are removed in an orderly fashion

so that components dependent on them are also removed. The procedure described for

the mutually exclusive relationship can be used to implement this operation in

accordance with the invention." See further 244-252, FIG. 6A)

In Appeal Brief, Appellant Argues:

> Claim 8
> Appellants incorporate herein, as also applying to claim 8, the arguments
> presented above with regard to claim 1. Additionally, Appellants note that
> independent claim 8 recites "determining from said semantic model, a set
> of dependent components required to be present in the specific target
> platform." The Examiner's analysis regarding these limitations is found on
> page 7 of the Fourth Office Action and reproduced below:
>
> determining from said semantic model (230), a set of dependent
> components required to be present in the specific target platform (232,
> see below Col. 14, Ln 13-31) (Col. 13, Ln 19-35 "Additional information
> may be received by the installation-affecting-relationship manager 220
> from a Core Engine 224, e.g., a list of components 222 to be installed and
> the like. The installation-affecting-relationship manager 220 computes
> entries for an installation-relationship matrix 226 and initializes parameters
> 228 and in a data structure 230 as well as an Installation_order list 232
> (step 202).") (emphasis in original)
>
> The Examiner's analysis is in error as to at least three issues. First, the
> passage cited by the Examiner does not describe that anything is obtained
> from the data structure 230 (i.e., "determining from said semantic model").
> Instead, the passage describes the manager 220 computing information to
> be included within the data structure 230.
>
> Second, the alleged teaching of "a set of dependent components" (i.e., the
> installation order list 232) is not derived from the data structure 230.
> Instead, the manager 220 creates the installation order list 232 while also
> creating the data structure 230.
>
> Third, the installation order list 232 is not described as a "a set of
> dependent components." Column 13, lines 33-35 of Marino specifically
> states the following: "there is no requirement that components 222 be
> related or be represented as a single tree or connected 2 graph." Since

there is no requirement that the components 222 be related, Marino
cannot teach 3 that the components are dependent components since a
dependency is a type of a relationship.

Examiner's Response:

Examiner respectfully disagrees. Appellant's arguments with respect to claim 8 are

similar to those presented for Claim 1, and have been addressed above.

Specifically, 1) Marino's further details the creation of the installation order list based on

the Relationship Matrix. (Col. 9, Ln 40-43) "The graph [stored in the matrix as described]

corresponding to the dependency relationships can be used to perform a depth first tree

traversal and generate a valid installation order, if one exists"…See further Col. 10, Ln

64-68: "Following an examination of the matrix in table 2, the component A3 is selected

and placed in an Installation order list because i[t] has an in-degree of 0 and has the

largest index among the nodes with in-degree of 0."

2) As described in Marino, the Installation-Relationship Matrix tracks dependencies

between components, which are used to generate the installation order. (Col. 9, Ln 40-

43) "The graph [stored in the matrix as described] corresponding to the dependency

relationships can be used to perform a depth first tree traversal and generate a valid

installation order, if one exists.")

3) the installation order list 232 is generated based on the relationship matrix (Col. 9, Ln

40-43) and therefore teaches "mapping said set of dependent components and said set

of resource requirements into platform specific instruction sin a platform specific

installation script."

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the

Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Matthew J. Brophy

Conferees:

/Lewis A. Bullock, Jr./
Supervisory Patent Examiner, Art Unit 2193

/Wei Y Zhen/
Supervisory Patent Examiner, Art Unit 2191